

APLICACIÓN DEL PROTOCOLO OGC® PUCK PARA INSTRUMENTACIÓN PLUG AND PLAYRS232 Y TCPIP

Luis VieiraSilva, Joaquin del Rio

**SARTI Research Group. Electronics Dept. Universitat Politècnica de Catalunya (UPC).
Rambla Exposició 24, 08800, Vilanova i la Geltrú. Barcelona. Spain.+(34) 938 967 200.
www.cdsarti.org**

Abstract

In the small market of precision measuring instruments there is no standardization of the communication protocols used for control and instrument configuration. Each manufacturer defines their own syntax and set of commands for each instrument. Due to this particular problem it is necessary to create a communication protocol that uses a common syntax and a set of common commands in addition to the specific commands of these instruments. Such protocol is the OGC® PUCK Protocol Standard that can be implemented in any measuring instrument or apparatus. This new protocol works through RS232 serial communication, most common for measuring instruments, but also for Ethernet communication (TCP-IP).

INTRODUCCIÓN

El principal objetivo es la implementación y posterior evaluación del protocolo sobre su usabilidad en instrumentos de medida. El protocolo OGC® PUCK Standard describe una interfaz de comunicación serie RS232 y Ethernet (TCP-IP) por la cual es posible identificar de manera única un dispositivo de medida, y el almacenaje de información respecto al dispositivo dentro del mismo. La implementación de dichas interfaces mediante la lenguaje de programación LabVIEW y el desarrollo de una API que facilite la generación de aplicaciones posteriores. Este proyecto viene a satisfacer una de las exigencias y necesidades del mercado industrial y de investigación en facilitar las comunicaciones entre el ordenador y el instrumento de medida, utilizando una plataforma visual plug&play (conectar y funcionar) donde sea mas fácil realizar: la instalación y configuración de forma automática o manual y mantenimiento de los instrumentos sin la necesidad de desplazarse al sitio donde se encuentra el instrumento. Una vez instalado y configurado el instrumento a través del "PuckMode" podrá proceder a la lectura y escrita de los valores obtenidos de los instrumentos solo cambiando en el programa para "Instrument Mode". A partir de este momento el instrumento está listo para realizar sus funciones de lectura y enviar los datos obtenidos a través de la misma puerta COM o TCPIP.

La configuración del instrumento en el ordenador no es solo una virtud de este proyecto pero también en acceder a sus datos tanto para su procesado, archivo o visualización compartida tanto en el instrumento como el ordenador. Para conseguir dicho objetivo vamos a dividir el problema en dos partes bien diferenciadas. Por un lado definiremos el nivel de instrumento y por otro lado, el nivel de gestión de datos con el usuario.

PROTOCOLO PUCK

PUCK (Programmable Underwater Connector, with Knowledge) es un protocolo de comandos simples que ayuda a automatizar el proceso de configuración, guardar información sobre el instrumento en el propio instrumento. La información almacenada (payload) puede ser relativa al instrumento (metadatos), código ejecutable o driver, o cualquier otra información que sea considerada necesaria para el sistema de observación. Cuando un instrumento PUCK-enable está conectado a un controlador, éste puede recuperar la información desde el instrumento a través del protocolo Puck.

El controlador podría instalar y ejecutar código relacionado con el instrumento, o incluir un datasheet electrónico (hoja de especificaciones) sobre el instrumento en un formato conocido (por ejemplo TEDS, Transducer Electronics DataSheet, de IEEE Std. 1451 o SensorML de OGC), o incluso información de cómo comunicar y configurar el instrumento (SID, Sensor Interface Descriptor). El proceso de configuración automática se refiere en conectar el instrumento y listo para trabajar mas conocido como plug&play o plug-and-work.

El protocolo PUCK no define cómo debe operar el instrumento, ni cómo debe controlarse sino cómo obtener información sobre el instrumento de una manera estándar. Cuando conectamos un instrumento Puck-enable a un equipo controlador, éste puede enviar comandos PUCK estándar que permiten almacenar o recuperar información del payload del instrumento.

La mayoría de instrumentos están diseñados para interactuar con un controlador a través de una interfaz serie. El controlador puede proporcionar una interfaz de usuario para el instrumento, registro de datos, o distribuir los datos a una red más amplia.

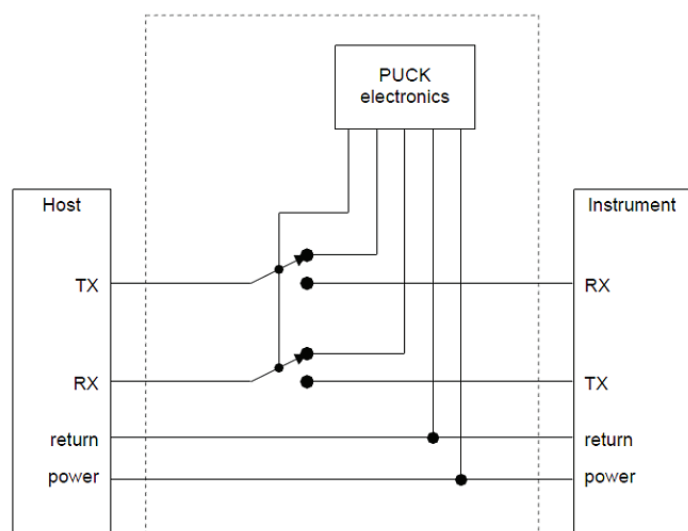


Figura 1. Cambio de PuckMode a Instrument Mode

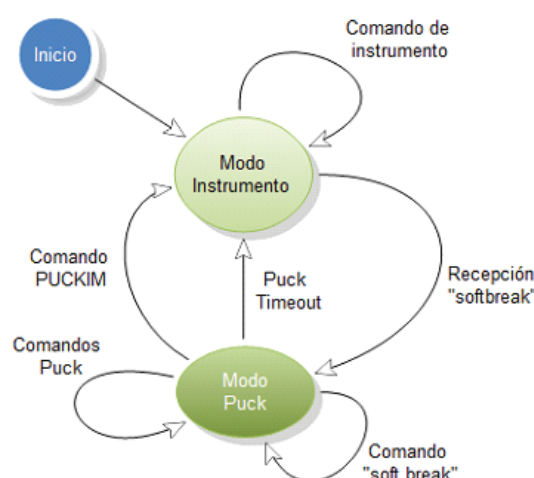


Figura 2. Cambio entre funcionamiento normal o en modo Puck

Con el propósito de realizar estas funciones, el controlador requiere información sobre el instrumento, tal como la configuración del puerto serie, conocimiento de los comandos reconocidos por el instrumento, y los metadatos que describen el instrumento y los datos científicos que genera. Los instrumentos que actualmente se comercializan no suministran esta información de forma automática, sino que requieren que el controlador sea configurado de antemano, y una parte del proceso de configuración generalmente consiste en la instalación del software o driver en el controlador quien gestiona la interacción con el instrumento. Esta configuración por lo general implica varios pasos, y puede ser un proceso largo, tedioso, y potencialmente propenso a errores. PUCK aborda esta cuestión de modo que permite la configuración automática del sistema cuando el instrumento está conectado al receptor. La lista de comandos que utiliza el protocolo Puck V1.4 son solo 13 y son los siguientes:

Comando	Descripción
PUCKRM	Lectura de la memoria PUCK
PUCKWM	Escritura de la memoria PUCK
PUCKFM	Finalizar sesión de escritura en la memoria PUCK
PUCKEM	Borrar la memoria PUCK
PUCKGA	Leer la dirección del puntero interno de memoria
PUCKIM	Configurar en modo instrumento
PUCKSA	Fijar la dirección del puntero interno de memoria
PUCKSZ	Leer el tamaño de memoria PUCK
PUCKSB	Fijar la velocidad de comunicación
PUCKTY	Leer el tipo de PUCK
PUCKVB	Verificar soporte a una determinada velocidad
PUCKVR	Leer la versión del protocolo PUCK
PUCK	Comando Nulo

Tabla 1. Comandos del protocolo PUCK

El mapa de memoria PUCK payload se define de la siguiente manera: los primeros 96 bytes (Instrument Datasheet) son reservados para la identificación del instrumento puede configurarse del tipo solo-lectura para asegurar los datos guardados no se altere accidentalmente durante toda la vida del instrumento y los restantes bytes hasta acabar la capacidad de la memoria es de uso libre. Es aconsejable dividir esa información por bloques "component #" en que la identificación de cada bloque en el mapa de memoria es dado por el "Payload tag".

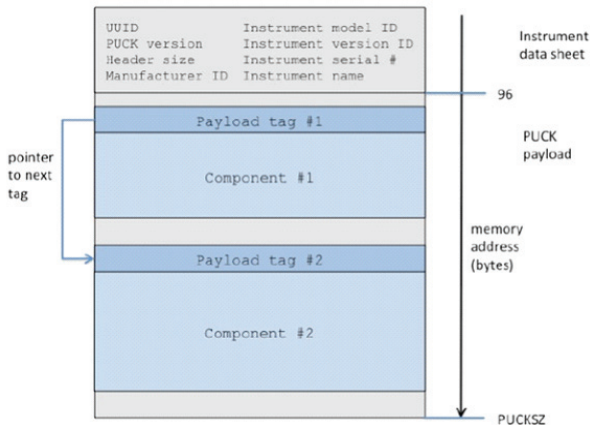


Figura 3. Mapa de memoria PUCK

El objetivo del payload es almacenar información sobre el instrumento, es decir, los metadatos del instrumento o también un driver ejecutable. En la tabla 2 podemos observar la descripción, el tamaño, y el formato de cada campo de los 96 bytes de identificación del instrumento.

Descripción	Tamaño (Bytes)	Formato
UUID del instrumento	16	UUID
Versión del datasheet del instrumento	2	U16
Tamaño del datasheet	2	U16
ID del fabricante	4	U32
Modelo del fabricante	2	U16
Versión del fabricante	2	U16
Numero de serie	4	U32
Nombre del instrumento	64	Char
TOTAL	96	

Tabla 2. Descripción del PUCK Instrument Datasheet del instrumento

OBJETIVO DEL TRABAJO

- La finalidad de este trabajo es conseguir implementar:
- La detección automática del puerto COM y velocidad Baudrate para el caso RS232 y buscar en una red TCP/IP el instrumento.
- Detección y alerta de conexión y desconexión de un instrumento PUCK.
- Notificación de la conexión o desconexión del instrumento.

En la figura 4 exponemos el algoritmo que implementa en la detección de conexión y desconexión de un instrumento PUCK. El programa envía periódicamente un PUCK soft-break a diferentes velocidades a cada uno de los puertos serie buscando un instrumento con el protocolo PUCK. Si se reconoce, el programa descarga el PUCK "instrument datasheet" y verifica si el instrumento conectado es un nuevo instrumento o no mediante su UUID (modelo del instrumento ID). Una vez inicializado el controlador recibirá los datos del instrumento periódicamente. Para identificar la desconexión del instrumento el algoritmo tiene en cuenta el periodo de lectura de datos del instrumento. Si el periodo de lectura es muy grande, por ejemplo superior a 2 segundos, o más, el programa verificará con una frecuencia mayor que el instrumento sigue conectado mediante un "Softbreak".

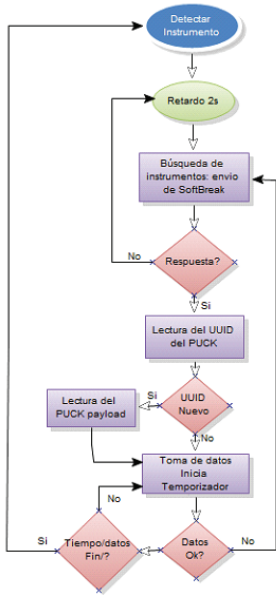


Figura 4. Algoritmo de conexión y desconexión de instrumentos PUCK.

DESCRIPCIÓN DE IP PUCK V1.4 TCP-IP

La novedad en la versión 1.4 en relación a la versión anterior es de poder comunicarse con el instrumento a través de una red TCP/IP. La implementación de IP PUCK implica comunicaciones a través de

la red Ethernet mediante protocolo TCP-IP. Para el establecimiento de una comunicación TCP-IP entre dos dispositivos o aplicaciones, la aplicación cliente deberá conocer la dirección IP de la aplicación servidor y además el puerto de comunicaciones donde dicha aplicación se mantiene a la espera de recibir peticiones de conexión. En el caso de IP PUCK, el puerto donde el instrumento que implemente IP PUCK espera la petición de una conexión por parte de una aplicación cliente que lo denominamos PUCK port.

El descubrimiento por parte de la aplicación cliente del valor del PUCK port se realiza a través del protocolo ZeroConf. El instrumento que implemente IP PUCK implementa también el protocolo ZeroConf para advertir su presencia en la red. Si el instrumento tiene comunicación Ethernet integrada en el propio instrumento como parte de su firmware, o a través de un interfaz programable serie-Ethernet puede realizar todas estas operaciones. Este programa tiene la ventaja de buscar automáticamente en la red dispositivos que implemente el protocolo Puck sin conocer su IP. Se conecta al primero que responde y muestra una lista de todos los instrumentos que hay conectados en la red.

DESARROLLO DEL PROGRAMA

Este programa fue realizado en LabVIEW de la National Instruments, con el objetivo de configurar el instrumento con el ordenador que tengan implementado el protocolo Puck: leer y escribir la identificación del instrumento, leer y escribir información libre en la memoria del instrumento a través de los Puck Payloads. También tiene la posibilidad de grabar esa información en un archivo en el ordenador u otro medio de memoria conectado al ordenador. Puede controlar el instrumento en Puckmode o Instrument Mode y modificar la identificación del instrumento. No obstante este programa no lee valores realizados con los instrumentos, solo realiza la configuración entre el ordenador y el instrumento.

El programa puede trabajar con instrumentos conectados en los puertos COM utilizando el protocolo RS232, Visa (National Instruments) u otro tipo de programa que pueda realizar un puente a una IP conectados a una red o simplemente conectado a una red TCP-IP.

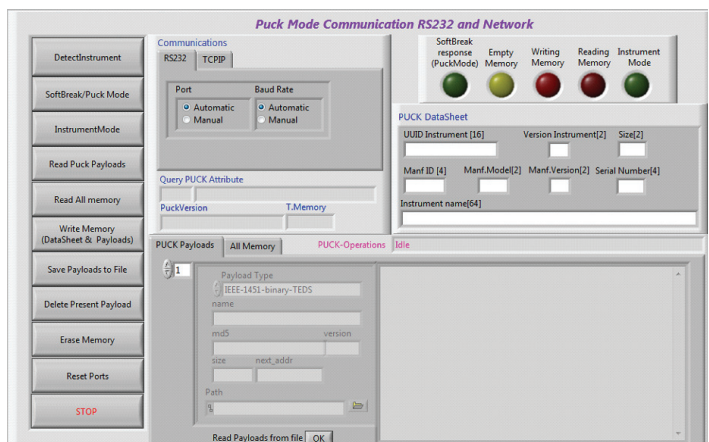


Figura 5. Ventana principal.

En RS232 el utilizador tiene la opción de escoger la detención automáticamente del puerto y la velocidad de trabajo (Baudrate) o escoger el automático de uno de ellos (puerto o velocidad). También tiene la opción de seleccionar los dos en manual. En la parte de TCPIP tenemos 2 opciones: Manual y automática. En manual es dedicada al Visa lo que deberá programar y activar la IP y puerto que está conectado el instrumento antes de ejecutar el programa. En automático el programa buscara en la rede instrumentos que implemente el protocolo Puck. Una vez ejecutado el programa el utilizador tiene la opción de escoger como encontrar el instrumento con RS232 o TCPIP. Luego deberá ejecutar el botón "Detect Instrument" y esperar que el programa detecte el instrument. Si lo encuentra, automáticamente lee los datos que se encuentra en la memoria: identificación del instrumento y todo que tenga en la memoria (Payloads). Es de subrayar, si hay más de un instrumento conectado en la red o en los puertos COM y la opción es de automático leerá los datos del primero instrumento que encuentra, puede que no sea el instrumento que se desea visualizar. Se así es, basta seleccionar el instrumento que desee trabajar a través de la lista

de instrumentos encontrados y volver a ejecutar el "Detect Instrument".



Figura 6. Lista instrumentos detectados

Este programa tiene la opción de cambiar de "Puck Mode" a "Instrument Mode" y viceversa, esto quiere decir que en el primero caso es para leer y programar el instrumento. Es necesario tener en cuenta que al ejecutar esta opción el instrumento deja de trabajar como instrumento, es decir deja de realizar y enviar las medidas que estaba ejecutando. En la segunda opción es para que el instrumento vuelva a ejecutar las funciones normales de funcionamiento.

Los siguientes botones son para leer el contenido de la memoria: por separado "Read Puck Payloads", "ReadAll Memory", toda la memoria en una lista incluyendo la identificación del instrumento.

El botón "Write Memory" es para escribir la información en el instrumento y el botón "Save Payloads to File" es para grabar toda la información del instrumento en un archivo del ordenador pero no graba al mismo tiempo en el instrumento deberá ejecutar el "Write Memory" si desea grabar en la memoria del instrumento.

El botón "Delete Present Payload" es para borrar el contenido actual del payload que esta visualizado y al mismo tiempo reorganiza todos los payloads.

El botón "Erase Memory" borra todo el contenido de la memoria incluyendo la identificación caso no este activada. El "Reset ports" es para buscar los puertos que están activos en el ordenador una vez ejecutado el programa, es función es solo para RS232 y Visa.

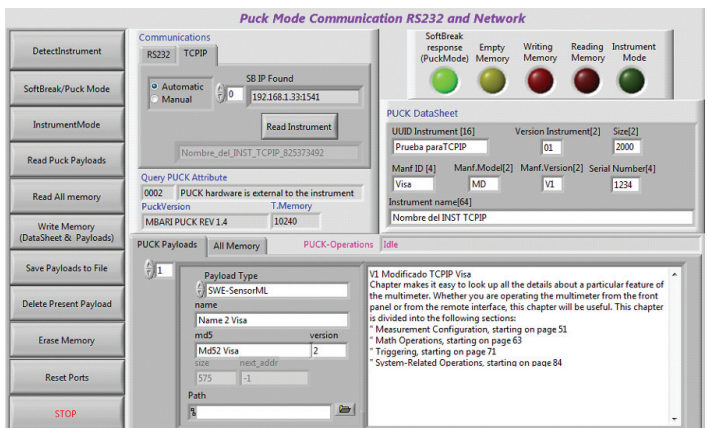


Figura 7. Ejemplo de una conexión TCPIP

REFERENCIAS

OGC® document: <http://www.opengis.net/spec/PUCK/v1.4/>
Reference number of this OGC® project document: OGC 09-127r2 Version: 1.4
Zero Configuration Networking: The Definitive Guide by Stuart Cheshire and Daniel H. Steinberg
www.ni.com
Tutor: Profesor Joaquin del Rio UPC - Vilanova i la geltru